Problem 3: Decryption Dilemma (15 points) [Email Submission]

By Vaibhav Mehta

RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission. The acronym "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977. In a public-key cryptosystem, the encryption key is public and distinct from the decryption key, which is kept secret (private). An RSA user creates and publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers are kept secret. Messages can be encrypted by anyone, via the public key, but can only be decoded by someone who knows the prime numbers. The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers.

Concretely, here is how the protocol works: Alice makes known two numbers, N and e which she has selected carefully. Together, these are her public key. Bob can use these numbers to encode a message and send it to Alice. A listener Eve knows N, e, and the encoded message. It is essentially impossible for Eve to decode the message, but Alice can decode the message easily because she knows the secret key. The heart of the RSA cryptosystem is the RSA "modulus" N. N is chosen to be a positive integer which equals the product of two distinct prime numbers p and q:

$$N = pq$$

We also need to pick the encoding exponent e. There are some mathematical constraints on e. In particular, e must be relatively prime to $(p-1)(q-1)$. Two integers are relatively prime if they share no common positive factors (divisors) except 1. Now with N and e chosen, A message m can be encoded as follows:

$$M = m^e \bmod N$$

M is known as the ciphertext. In order to decrypt her message, Alice must also compute a decryption exponent d. You do not need to know the math behind decryption to solve the problems, but if you are interested, see here.

1 Part 1 (5 Points)

Excited by your new-found knowledge of cryptography, you apply to a cryptocurrency firm, Cr33pt0. Unfortunately, the security people at Cr33pt0 are not as competent as you. One day, while poking around, you discover a file with a public key, and an encrypted message. Curious, as you are, you try to analyze this with your new-found skills. We have given some Python code to help you get started. Your solution does not have to be in Python, but the starter code might provide some useful strategies. You may assume, that the security is weak, including that there is no padding. Decrypt the message!

2 Part 2 (10 Points)

Your boss finds out about what you have been doing. So the company implements a new security protocol, which they don't tell you about. However, one day, you find a screenshot on your boss' computer. Your boss used a program to encrypt their secret message. They then redacted the keys and their first message. They forgot to redact the second message. Can you use this to find their first message?

Hint: Poking around has gotten you this far. Poking around with the image might save you a lot of typing.

Find the first message!

Your Boss has used Pycryptodome, a very useful Python library for cryptography. You can read the docs here. Also note the use of padding. This is done for added security. You can still use any language of your choosing with its own crypto library. In case you cannot figure out the details, but you have a fair idea of how one might do this, you send a write-up as part of your submission for partial credit.

How to Submit

Email each part separately to coscon.submit@gmail.com. If you must resubmit, *respond to the thread where you sent your original submission; we cannot guarantee that your resubmission will be graded otherwise.*

Part 1
Store your decrypted message in a text file called *Problem3aMessage.txt*. Send an email with *exact* subject *Problem3aSubmission* and this file as an attachment. Also attach any code that you wrote to decrypt the message.

Part 2
Store the decrypted first message in a text file called *Problem3bMessage.txt*. If you are including a write-up for partial credit, then save this in a pdf titled *Problem3bWriteup.pdf*. If you have successfully decrypted the first message, then the writeup is optional. Send an email with *exact* subject *Problem3bSubmission* and these files as an attachment. Attach any code that you wrote to decrypt the message.